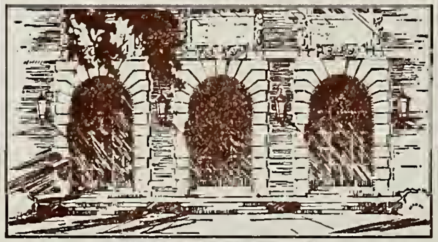

ON FINDING THE MAXIMAL ELEMENTS IN A SET OF PLANE VECTORS

by

Foong Frances Yao

July, 1974

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN · URBANA, ILLINOIS

UIUCDCS-R-74-667

ON FINDING THE MAXIMAL ELEMENTS IN A SET OF PLANE VECTORS

By

Foong Frances Yao

July, 1974

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN
URBANA, ILLINOIS 61801

*Supported in part by the National Science Foundation under contract NSF GJ 41538.

Abstract

Let F be a set of n vectors in the plane. A partial order is defined on F in a natural manner. It is known that the maximal elements of F can be found in S(n) + n-1 comparisons, where S(n) is the minimum number of comparisons required to sort n numbers. In this note we show that S(n) + n-1 comparisons are necessary.

1. Introduction

Let $F = \{(x_i,y_i) | \ i=1,2,\ldots,n\}$ be a set of n distinct vectors in the plane. A vector $(x_i y_i) \varepsilon F$ is said to be an maximal element of F if for any $(x_j,y_j) \varepsilon F$ where $1 \leq j \leq n$ and $j \neq i$, we have either $x_j < x_i$ or $y_j < y_i$. We will use $F^{(max)}$ to denote the set of maximal elements of F.

As noted by Luccio and Preparata [1], it is possible to find $F^{(max)}$ by using no more than S(n)+n-1 pairwise comparisons among the numbers $\{x_1,x_2,\ldots,x_n,y_1,y_2,\ldots,y_n\}$, where S(n) is the minimax number of comparisons for sorting n numbers. If we denote by V(n) the minimum number of comparisons required to find $F^{(max)}$ for any set F of n vectors, we will show that

$$V(n) \geq S(n) + n-1 \qquad (1)$$

In fact we will prove that S(n) + n-1 comparisons are necessary even for algorithms whose input is restricted to those F's satisfying $x_i > y_j$ for all $1 \leq i, j \leq n$. Under this restriction, we can assume that the algorithms to be considered contain only comparisons of the form $x_i:x_j$ or of the form $y_k:y_\ell$. Let $\mathcal{A}$ be any such algorithm. If $T_{\mathcal{A}}$ is the decision of $\mathcal{A}$, we will show that some subtree $T'_{\mathcal{A}}$ of $T_{\mathcal{A}}$ is isomorphic to the decision tree $T_{\mathcal{S}}$ of a sorting algorithm $\mathcal{S}$ (for n numbers). This implies that $V(n) \geq S(n)$. The algorithm $\mathcal{S}$, however, is not an optimal sorting algorithm. In fact, the height of $T_{\mathcal{S}}$ can be reduced by n-1 when some redundant comparisons are removed. This then leads to

$$V(n) \geq S(n) + n-1.$$

Details of the above scheme of proof are given in the next two sections.

2. Definition of $T'_{\mathcal{A}}$

Let us consider those sets of vectors $F = \{(x_i, y_i) | 1 \leq i \leq n\}$ with the property

$$x_i > x_j \text{ iff } y_i < y_j \qquad \text{for all } i,j. \qquad (2)$$

For F satisfying (2), all n vectors are maximal elements. The following lemma is essential to the proof of (1) in the next section.

<u>Lemma</u>| Let $\mathcal{A}$ be an algorithm for finding maximal vectors. If F satisfies (2), then there exists a permutation $(i_1, i_2, \ldots, i_n)$ of $(1,2,\ldots,n)$ such that when $\mathcal{A}$ is applied to F, the following statements are ture:

(i) Algorithm $\mathcal{A}$ establishes

$$x_{i_1} > x_{i_2} > \ldots > x_{i_n} \quad \text{and} \quad y_{i_1} < y_{i_2} < \ldots < y_{i_n} \qquad (3)$$

(ii) The following comparisons are made:

$$\begin{array}{llll} x_{i_1}:x_{i_2} & x_{i_2}:x_{i_3} & \cdots & x_{i_{n-1}}:x_{i_n} \\ y_{i_1}:y_{i_2} & y_{i_2}:y_{i_3} & \cdots & y_{i_{n-1}}:y_{i_n} \end{array} \qquad (4)$$

<u>Proof</u>: Since every vector of F is an maximal element, algorithm $\mathcal{A}$ must establish, for any $i \neq j$, either $(x_i > x_j) \wedge (y_i < y_j)$ or $(x_i < x_j) \wedge (y_i > y_j)$. Therefore (3) is true. All the comparisons in (4) have to be made since they are necessary for establishing (3). □

We now turn to the definition of $T'_{\mathcal{A}}$ as mentioned in Sec. 1. Let $\mathcal{A}$ be an algorithm for finding maximal vectors and $T_{\mathcal{A}}$ its decision tree. For any input F, there is a unique path in $T_{\mathcal{A}}$ which determines the actual computing process when $\mathcal{A}$ is applied to F. We shall say it is the path <u>traversed</u> by F.

Definition 1 For any algorithm $\mathcal{A}$ that finds maximal vectors, $T'_{\mathcal{A}}$ is defined to be the subtree of $T_{\mathcal{A}}$ consisting of all those paths traversed by the F's satisfying (2).

3. Constructing a sorting algorithm from $T'_{\mathcal{A}}$

Let $T'_{\mathcal{A}}$ be the subtree obtained from $T_{\mathcal{A}}$ as in Definition 1. We will transform $T'_{\mathcal{A}}$ into the decision tree $T_{\mathcal{S}}$ for an algorithm $\mathcal{S}$ which sorts n numbers.

Definition 2 Let $\{z_1, z_2, \ldots, z_n\}$ represent n distinct numbers. We define a new decision tree $T_{\mathcal{S}}$ based on $T'_{\mathcal{A}}$ as follows:

(i) Replace any comparison of the form $x_i : x_j$ at an internal node of $T'_{\mathcal{A}}$ with $z_i : z_j$. Also replace the branching labels $x_i > x_j$ and $x_i < x_j$ on the arcs with $z_i > z_j$ and $z_i < z_j$ respectively.

(ii) Replace any comparison $y_i : y_j$ by $z_i : z_j$. However, the branching label $y_i > y_j$ is replaced by $z_i < z_j$ while $y_i < y_j$ is replaced by $z_i > z_j$.

(iii) Leave the external nodes blank at present.

We will show that the tree $T_{\mathcal{S}}$ so obtained indeed represents a sorting algorithm. But first note that the tree structures of $T_{\mathcal{S}}$ and $T'_{\mathcal{A}}$ are isomorphic in a natural way. Let us denote by α this isomorphic mapping from $T'_{\mathcal{A}}$ onto $T_{\mathcal{S}}$. If N is a node performing $x_i : x_j$ in $T'_{\mathcal{A}}$, then $\alpha(N)$ is a node in $T_{\mathcal{S}}$ performing $z_i : z_j$. Similarly if C is an arc in $T'_{\mathcal{A}}$ with branching label $y_i > y_j$, then $\alpha(C)$ is an arc in $T_{\mathcal{S}}$ with branching label $z_i < z_j$. For a set P of nodes and arcs in $T'_{\mathcal{A}}$, we shall also use $\alpha(P)$ to denote the set of corresponding nodes and arcs in $T_{\mathcal{S}}$.

The following lemma is obvious from the definition of $T_{\mathcal{S}}$.

Lemma 2

Let $Z = \{z_i, z_2, \ldots, z_n\}$ be a set of n distinct numbers, and $F = \{(x_i, y_i) |\ i = 1,2,\ldots,n\}$ be a set of vectors satisfying (2). Moreover, assume that

$$x_i > x_j\ ,\ y_i < y_j \quad \text{iff } z_i > z_j \quad \text{for all } i,j. \tag{5}$$

Then the path P traversed by Z in $T_{\mathcal{S}}$ corresponds to the path Q traversed by F in $T'_{\mathcal{A}}$ in the sense that $P = \alpha(Q)$.

Lemma 2 implies that, if $x_i{:}x_j$ (or $y_i{:}y_j$) is performed when $\mathcal{A}$ is applied to F, then $z_i{:}z_j$ is performed when $\mathcal{S}$ is applied to Z for F,Z satisfying (5).

We are now ready to prove that $\mathcal{S}$ is a sorting algorithm.

Theorem 1

(i) $\mathcal{S}$ is a sorting algorithm for $Z = \{z_1, z_2, \ldots, z_n\}$.

(ii) For any input Z, there are n-1 comparisons each of which is performed twice in $\mathcal{S}$.

Proof: Consider any set of n distinct numbers $Z = \{z_1, z_2, \ldots, z_n\}$. Assume $z_{i_1} > z_{i_2} > \ldots > z_{i_n}$. Now consider the following set of vectors:

$$F = \{(x_i, y_i) |\ 1 \le i \le n,\ \text{where } x_{i_1} > x_{i_2} > \ldots > x_{i_n} \text{ and } y_{i_1} < y_{i_2} < \ldots < y_{i_n}.\}$$

When $\mathcal{A}$ is applied to F, the comparisons

$$\begin{array}{llll} x_{i_1}{:}x_{i_2} & x_{i_2}{:}x_{i_3} & \cdots & x_{i_{n-1}}{:}x_{i_n} \\ & & & \\ y_{i_1}{:}y_{i_2} & y_{i_2}{:}y_{i_3} & \cdots & y_{i_{n-1}}{:}y_{i_n} \end{array} \tag{6}$$

are performed according to Lemma 1. Therefore, when $\mathcal{S}$ is applied to Z, each of the n-1 comparisons

$$z_{i_1}:z_{i_2} \quad z_{i_2}:z_{i_3} \quad \cdots \quad z_{i_{n-1}}:z_{i_n} \tag{7}$$

will be performed twice (duplicate images of $x_{i_k}:x_{i_{k+1}}$ and $y_{i_k}:y_{i_{k+1}}$ under mapping α) by Lemma 2 and (6). Since the comparisons in (7) suffice to extablish $z_{i_1}>z_{i_2}>\ldots>z_{i_n}$, we have sorted Z. $\square$

As a result of Theorem 1, we can clearly remove the redundant comparisons from $\mathcal{S}$ to obtain a sorting algorithm which makes n-1 fewer comparisons than $\mathcal{S}$ for any input $Z = \{z_1,z_2,\ldots,z_n\}$. This shows that the height $h_{\mathcal{S}}$ of $T_{\mathcal{S}}$ satisfies

$$h_{\mathcal{S}} \geq S(n) + n-1.$$

On the other hand, since $T_{\mathcal{S}}$ is isormorphic to a subtree of $T_{\mathcal{A}}$, the height $h_{\mathcal{A}}$ of $T_{\mathcal{A}}$ must then satisfy

$$h_{\mathcal{A}} \geq S(n) + n-1. \tag{8}$$

Since (8) is true for any algorithm $\mathcal{A}$ that finds the maximal vectors, we thus obtain our main result:

<u>Theorem 2</u> $V(n) \geq S(n) + n-1$

As mentioned in Sec. 1, $S(n) + n-1$ is an upper bound for $V(n)$ since $F^{(max)}$ can be found by sorting the vectors of F into non-increasing order by their first coordinates, and then making a sequential search on their second coordinates. Therefore we have $V(n) = S(n) + n-1$.

Acknowledgement

H.T. Kung has also considered this problem independently in [2] and obtained a weaker result.

REFERENCES

[1] Luccio, F., and Preparata, F.P., On Finding the Maxima of a Set of Vectors, Instituto di Scienze dell'Informazione, Università di Pisa, Corso Italia 40, 56100 Pisa, Italy, 1973.

[2] Kung, H.T., On the Computational Complexity of Finding the Maxima of a Set of Vectors, 15th Annual Symposium of SWAT, October, 1974.

BIBLIOGRAPHIC DATA SHEET	1. Report No. UIUCDCS-R-74-667	2.	3. Recipient's Accession No.

4. Title and Subtitle
On Finding the Maximal Elements in a Set of Plane Vectors

5. Report Date
July, 1974

6.

7. Author(s)
Foong Frances Yao

8. Performing Organization Rept. No. UIUCDCS-R-74-667

9. Performing Organization Name and Address
University of Illinois
Department of Computer Science
Urbana, IL 61801

10. Project/Task/Work Unit No.

11. Contract/Grant No.
NSF GJ 41538

12. Sponsoring Organization Name and Address
National Science Foundation
1800 G St. N.W.
Washington, D.C. 20550

13. Type of Report & Period Covered

14.

15. Supplementary Notes

16. Abstracts

Let F be a set of n vectors in the plane. A partial order is defined on F in a natural manner. It is known that the maximal elements of F can be found in S(n) + n-1 comparisons, where S(n) is the minimum number of comparisons required to sort n numbers. In this note we show that S(n) + n-1 comparisons are necessary.

17. Key Words and Document Analysis. 17a. Descriptors

Maximal element, decision tree

17b. Identifiers/Open-Ended Terms

17c. COSATI Field/Group

18. Availability Statement	19. Security Class (This Report)	21. No. of Pages
Unlimited	UNCLASSIFIED	10
	20. Security Class (This Page) UNCLASSIFIED	22. Price